

TITLE

COMPUTER SYSTEM PROVIDED WITH HOTKEYS

BACKGROUND OF THE INVENTION

Field of the Invention:

5 The present invention relates to a computer system provided with hotkeys and particularly to a computer system capable of implementing hotkey functions.

Description of the Prior Art:

FIG. 1 is a diagram showing a computer system provided
10 with hotkeys. The computer system 100 of FIG. 1 includes a CPU 102 coupled to a bridge logic device 106 via a CPU bus. The bridge logic device 106 is sometimes referred to as a "North bridge" for no other reason than it often is depicted at the upper end of a computer system drawing. The North
15 bridge 106 also couples to a main memory array 104 by a memory bus, and may further couple to a graphics controller 108 via an accelerated graphics port (AGP) bus. The North bridge 106 couples CPU 102, memory 104, and graphics controller 108 to the other peripheral devices in the system
20 through a primary expansion bus (BUS A) which may be implemented as a peripheral component interconnect (PCI) bus or an extended industry standard architecture (EISA) bus. Various components that comply with the communications protocol and electrical requirements of BUS A may reside on
25 this bus, such as an audio device 114, an IEEE 1394 interface device 116, and a network interface card (NIC) 118. The system may include more than one network interface, as indicated by NIC 119. These components may be integrated onto the motherboard or they may be plugged into
30 expansion slots 110 that are connected to BUS A.

If other secondary expansion buses are provided in the computer system, as is typically the case, another bridge logic device 112 is used to couple the primary expansion bus (BUS A) to the secondary expansion bus (BUS B). Various components that comply with the communications protocol and electrical requirements of BUS B may reside on this bus, such as a hard disk controller 122, Flash ROM 124, and Super I/O controller 126. Slots 120 may also be provided for plug-in components that comply with the protocol of BUS B. Flash ROM 124 stores the system BIOS that is executed by CPU 102 during system initialization.

Super I/O controller 126 may include battery-backed CMOS memory for storing BIOS configuration parameters for system 100, and may further include a counter or a Real Time Clock (RTC). The RTC may be used to track the activities of certain components such as the hard disk 122 and the primary expansion bus, so that controller 126 can induce a sleep mode or reduced power mode after a predetermined time of inactivity. In order to support hotkey functions, the super I/O controller 126 may also include an embedded controller to detect the hotkey events and initiate a procedure for implementation of the hotkey functions.

FIG. 2 shows a conventional embedded controller. The embedded controller 1261 is disposed in the super I/O controller 126 and coupled between the keyboard 130 and south bridge 112. For example, when the keys Fn and F2 are pressed together, the keyboard 130 sends to the input terminals KBD_IN and GPE_IN of the embedded controller 1261 a corresponding scan code and a signal representing an interrupt event through the output terminals KBD_O and

Fn_SMI respectively. The keyboard 130 is typically composed of a microprocessor, decoder, and a matrix of key switches. The microprocessor periodically monitors the matrix to detect whether a key is pressed. The address of the pressed key in the matrix is decoded by the decoder for generation of the corresponding scan code. Additionally, a logic circuit, such as an AND gate, detects whether a function key is pressed together with another key. When a function key is pressed together with another key, the logic circuit asserts a signal representing an interrupt event. The embedded controller 1261 transfers the scan code from the keyboard 130 to the south bridge 112 and asserts a signal input to the input terminal EXT_SMI of the south bridge 112 to initiate the procedure for the hotkey function. The procedure is implemented by the ASL code (ACPI source language code) stored in the BIOS.

FIG. 3 is a flowchart of a conventional method for implementing hotkey functions in a computer.

In step 31, the key Fn and F2 are pressed together so that a hotkey event occurs. The keyboard accordingly sends out corresponding scan codes and asserts the signal representing an interrupt event on the output terminals KBD_O and Fn_SMI respectively.

In step 32, the operating system stored in the hard disk 122 has been loaded into the CPU 100. The signal representing the interrupt event from the keyboard 130 results in an SMI command sent to the CPU 100, which temporarily stops all procedures currently implemented by the operating system.

In step 33, in cooperation with the embedded controller 1261, the CPU 100 loads the ASL code from the BIOS and implements the procedure thereof.

5 In step 34, the CPU 100 determines whether the interrupt event is a hotkey event. If so, the procedure proceeds to step 35; otherwise, the procedure of the ASL code is terminated and the CPU 100 returns to continue the procedures of the operating system previously stopped by the SMI command.

10 In step 35, the CPU 100 identifies the type of operating system. The procedure proceeds to step 361 if the operating system is Microsoft Windows 98 or ME, and proceeds to step 362 if the operating system is Microsoft Windows 2000 or XP.

15 In step 361, the scan code from the keyboard 130 is transferred to a hotkey driver identifying a corresponding status value and driving a corresponding device to implement the hotkey function.

20 In step 362, the embedded controller 1261 identifies a corresponding status value and driving a corresponding device to implement the hotkey function.

However, in the conventional computer system provided with hotkeys, the embedded controller increases the cost and complicates the circuitry of the super I/O controller 126.
25 Further, the ASL code must be adapted to the embedded controller, which is troublesome in ASL coding.

SUMMARY OF THE INVENTION

The present invention provides a computer system capable of implementing hotkey functions.

The present invention provides a method for implementing functions of hotkeys in a computer having a BIOS and operating system, the method comprising the steps of asserting a system management interrupt signal when a
5 system management interrupt event is detected, determining whether the system management interrupt event results from one of the hotkeys being pressed, determining whether the operating system is of a first or second type if the system management interrupt event results from one of the hotkey
10 being pressed, reading a status value corresponding to the pressed hotkey and implementing one of the functions corresponding to the status value by the BIOS if the operating system is of the first type, and reading the status value corresponding to the pressed hotkey and
15 transferring the status value to the operating system by the BIOS, and implementing one of the functions corresponding to the status value by the operating system if the operating system is of the second type.

The present invention further provides a computer
20 capable of implementing the functions of hotkeys comprising a memory device storing an operating system, a BIOS storing program codes and status values, a keyboard having a first and second key, and asserting a first and second key press signal when the first and second keys are pressed
25 respectively, a bridge device asserting an interrupt signal representing a hotkey event when the first and second key press signal are asserted simultaneously, and a CPU loaded with the operating system, and when the interrupt signal is asserted, reading the program codes in the BIOS to implement
30 the steps of asserting an SMI signal, determining whether

the interrupt signal results from the hotkey event,
determining whether the operating system is of a first or
second type if the interrupt signal results from the hotkey
event, reading one of the status values corresponding to the
5 hotkey event and implementing one of the functions
corresponding to the read status value by the BIOS if the
operating system is of the first type, and reading one of
the status values corresponding to the hotkey event and
transferring the read status value to the operating system
10 by the BIOS, and implementing one of the functions
corresponding to the read status value by the operating
system if the operating system is of the second type.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention will become more fully understood
15 from the detailed description given hereinbelow and the
accompanying drawings, given by way of illustration only and
thus not intended to be limitative of the present invention.

FIG. 1 is a diagram showing a conventional computer
system provided with hotkeys.

20 FIG. 2 shows a conventional embedded controller.

FIG. 3 is a flowchart of a conventional method for
implementing hotkey functions in a computer.

FIG. 4 is a diagram showing a computer capable of
implementing functions of hotkeys according to one
25 embodiment of the invention.

FIG. 5 is a flowchart of a method for implementing
functions of hotkeys in a computer according to one
embodiment of the invention.

DETAILED DESCRIPTION OF THE INVENTION

FIG. 4 is a diagram showing a computer capable of implementing functions of hotkeys according to one embodiment of the invention. It is similar to that shown in FIG. 1 except that the I/O controller does not include an embedded controller. For example, when the keys Fn and F2 are pressed together, the keyboard 130 sends directly to the south bridge 112 a corresponding scan code and a signal representing an interrupt event through the output terminals KBD_O and Fn_SMI respectively, by which the CPU 100 loads the ASL code from the BIOS initiating the procedure for the hotkey function.

FIG. 5 is a flowchart of a method for implementing functions of hotkeys in a computer according to one embodiment of the invention.

In step 51, the keys Fn and F2 are pressed together so that a hotkey event occurs. The keyboard accordingly sends out corresponding scan codes and asserts a signal representing an interrupt event.

In step 52, the operating system stored in the hard disk 122 has been loaded into the CPU 100. The signal representing the interrupt event from the keyboard 130 results in an SMI command sent to the CPU 100, which temporarily stops all procedures currently implemented by the operating system.

In step 53, the CPU 100 loads the ASL code from the BIOS and accordingly implements the following steps.

In step 54, the CPU 100 determines whether the interrupt event is a hotkey event. If so, the procedure proceeds to step 55; otherwise, the procedure of the ASL

code is terminated and the CPU 100 returns to continue the procedures of the operating system previously stopped by the SMI command.

5 In step 55, the CPU 100 identifies the type of the operating system. The procedure proceeds to step 561 if the operating system is Microsoft Windows 98 or ME, and proceeds to step 562 if the operating system is Microsoft Windows 2000 or XP.

10 In step 561, the CPU 100 reads the scan code from the keyboard 130 identifies a corresponding status value stored in the BIOS, transfers the status value to a corresponding driving device implementing the hotkey function.

15 In step 562, the CPU 100 reads the scan code from the keyboard 130 identifies a corresponding status value stored in the BIOS, transfers the status value to the operating system to drive a corresponding device implementing the hotkey function.

20 In conclusion, the present invention provides a computer system capable of implementing hotkey functions without using an embedded controller. The ASL code stored in the BIOS, instead of an embedded controller, identifies hotkey events and initiates the procedure for implementing hotkey functions, which eliminates the need for a hotkey driver, the additional cost and circuit complexity resulting from the embedded controller, and simplifies the ASL coding.

25 The foregoing description of the preferred embodiments of this invention has been presented for purposes of illustration and description. Obvious modifications or variations are possible in light of the above teaching. The 30 embodiments were chosen and described to provide the best

Client Ref.:VIT02-0195US
Our ref:0608-8617-US/final/Vincent/Steve

illustration of the principles of this invention and its
practical application to thereby enable those skilled in the
art to utilize the invention in various embodiments and with
various modifications as are suited to the particular use
5 contemplated. All such modifications and variations are
within the scope of the present invention as determined by
the appended claims when interpreted in accordance with the
breadth to which they are fairly, legally, and equitably
entitled.

10